

Expectation-maximization Gaussian-mixture Approximate Message Passing

Jeremy P. Vila and Philip Schniter*

Abstract—When recovering a sparse signal from noisy compressive linear measurements, the distribution of the signal’s non-zero coefficients can have a profound affect on recovery mean-squared error (MSE). If this distribution was apriori known, then one could use computationally efficient approximate message passing (AMP) techniques for nearly minimum MSE (MMSE) recovery. In practice, though, the distribution is unknown, motivating the use of robust algorithms like Lasso—which is nearly minimax optimal—at the cost of significantly larger MSE for non-least-favorable distributions. As an alternative, we propose an empirical-Bayesian technique that simultaneously learns the signal distribution while MMSE-recovering the signal—according to the learned distribution—using AMP. In particular, we model the non-zero distribution as a Gaussian mixture, and learn its parameters through expectation maximization, using AMP to implement the expectation step. Numerical experiments on a wide range of signal classes confirm the state-of-the-art performance of our approach, in both reconstruction error and runtime, in the high-dimensional regime.

I. INTRODUCTION

We consider estimating a K -sparse (or compressible) signal $\mathbf{x} \in \mathbb{R}^N$ from $M < N$ linear measurements $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \in \mathbb{R}^M$, where \mathbf{A} is known and \mathbf{w} is additive white Gaussian noise (AWGN). For this problem, accurate (relative to the noise variance) signal recovery is known to be possible with polynomial-complexity algorithms when \mathbf{x} is sufficiently sparse and when \mathbf{A} satisfies certain restricted isometry properties [4], or when \mathbf{A} is large with i.i.d random entries [5] as discussed below.

Lasso [6] (or, equivalently, Basis Pursuit Denoising [7]), is a well-known approach to the sparse-signal recovery problem that solves the convex problem

$$\hat{\mathbf{x}}_{\text{lasso}} = \arg \min_{\hat{\mathbf{x}}} \|\mathbf{y} - \mathbf{A}\hat{\mathbf{x}}\|_2^2 + \lambda_{\text{lasso}} \|\hat{\mathbf{x}}\|_1, \quad (1)$$

with λ_{lasso} a tuning parameter that trades between the sparsity and measurement-fidelity of the solution. When \mathbf{A} is constructed from i.i.d entries, the performance of Lasso can be sharply characterized in the large system limit (i.e., as $K, M, N \rightarrow \infty$ with fixed undersampling ratio M/N and sparsity ratio K/M) using the so-called phase transition curve

The authors are with the Department of Electrical and Computer Engineering, The Ohio State University, Columbus, OH.

*Please direct all correspondence to Prof. Philip Schniter, Dept. ECE, The Ohio State University, 2015 Neil Ave., Columbus OH 43210, e-mail: schniter@ece.osu.edu, phone 614.247.6488, fax 614.292.7596.

This work has been supported in part by NSF-IUCRC grant IIP-0968910, by NSF grant CCF-1018368, and by DARPA/ONR grant N66001-10-1-4090.

Portions of this work were presented at the Duke Workshop on Sensing and Analysis of High-Dimensional Data in July 2011 [1]; the Asilomar Conference on Signals, Systems, and Computers in Nov. 2011 [2]; and the Conference on Information Science and Systems in Mar. 2012 [3].

(PTC) [5], [8]. When the observations are noiseless, the PTC bisects the M/N -versus- K/M plane into the region where Lasso reconstructs the signal perfectly (with high probability) and the region where it does not. (See Figs. 2-4.) When the observations are noisy, the same PTC bisects the plane into the regions where Lasso’s noise sensitivity (i.e., the ratio of estimation-error power to measurement-noise power under the worst-case signal distribution) is either finite or infinite [9]. An important fact about Lasso’s noiseless PTC is that it is invariant to the distribution of the nonzero signal coefficients. In other words, if the vector \mathbf{x} is drawn i.i.d from the pdf

$$p_X(\mathbf{x}) = \lambda f_X(\mathbf{x}) + (1 - \lambda)\delta(\mathbf{x}), \quad (2)$$

where $\delta(\cdot)$ is the Dirac delta, $f_X(\cdot)$ is the active-coefficient pdf (with zero probability mass at $\mathbf{x} = 0$), and $\lambda \triangleq K/N$, then the Lasso PTC is invariant to $f_X(\cdot)$. While this implies that Lasso is robust to “difficult” instances of $f_X(\cdot)$, it also implies that Lasso cannot benefit from the case that $f_X(\cdot)$ is an “easy” distribution. For example, when the signal is known apriori to be nonnegative, polynomial-complexity algorithms exist with PTCs that are better than Lasso’s [10].

At the other end of the spectrum is minimum mean-squared error (MMSE)-optimal signal recovery under *known* marginal pdfs of the form (2) and *known* noise variance. The PTC of MMSE recovery has been recently characterized [11] and shown to be well above that of Lasso. In particular, for *any* $f_X(\cdot)$, the PTC on the M/N -versus- K/M plane reduces to the line $K/M = 1$ in both the noiseless and noisy cases. Moreover, efficient algorithms for approximate MMSE-recovery have been proposed, such as the Bayesian version of Donoho, Maleki, and Montanari’s *approximate message passing* (AMP) algorithm from [12], which performs loopy belief-propagation on the underlying factor graph using central-limit-theorem approximations that become exact in the large-system limit under i.i.d \mathbf{A} . Although AMP’s complexity is remarkably low (e.g., dominated by one application of \mathbf{A} and \mathbf{A}^T per iteration with typically < 50 iterations to convergence), it offers rigorous performance guarantees in the large-system limit [13]. To handle arbitrary noise distributions and a wider class of matrices \mathbf{A} , Rangan proposed a *generalized AMP* (GAMP) [14] that forms the starting point of this work. (See Table I.)

In practice, one ideally wants a recovery algorithm that does not need to know $p_X(\cdot)$ and the noise variance a priori, yet offers performance on par with MMSE recovery, which (by definition) requires knowing these prior statistics. Towards this goal, we propose a recovery scheme that aims to *learn* the prior signal distribution $p_X(\cdot)$, as well as the variance of the AWGN,

while simultaneously recovering the signal vector \mathbf{x} from the noisy compressed measurements \mathbf{y} . To do so, we model the active component $f_X(\cdot)$ in (2) using a generic L -term Gaussian mixture (GM) and then learn the GM parameters and noise variance using the expectation-maximization (EM) algorithm [15]. As we will see, all of the quantities needed for the EM updates are already computed by the GAMP algorithm, making the overall process very computationally efficient. Moreover, GAMP provides approximately MMSE estimates of \mathbf{x} that suffice for signal recovery, as well as posterior activity probabilities that suffice for support recovery.

Since, in our approach, the prior pdf parameters are treated as deterministic unknowns, our proposed EM-GM-AMP algorithm can be classified as an “empirical-Bayesian” approach [16]. Compared with previously proposed empirical-Bayesian approaches to compressive sensing (e.g., [17]–[19]), ours has a more flexible signal model, and thus is able to better match a wide range of signal pdfs $p_X(\cdot)$, as we demonstrate through a detailed numerical study. In addition, the complexity scaling of our algorithm is superior to that in [17]–[19], implying lower complexity in the high dimensional regime, as we confirm numerically.

Notation: For matrices, we use boldface capital letters like \mathbf{A} , and we use $\text{tr}(\mathbf{A})$ and $\|\mathbf{A}\|_F$ to denote the trace and Frobenius norm, respectively. Moreover, we use $(\cdot)^\top$ to denote transpose, $(\cdot)^*$ conjugate, and $(\cdot)^H$ conjugate transpose. For vectors, we use boldface small letters like \mathbf{x} , and we use $\|\mathbf{x}\|_p = (\sum_n |x_n|^p)^{1/p}$ to denote the ℓ_p norm, with x_n representing the n^{th} element of \mathbf{x} . For a Gaussian random vector \mathbf{x} with mean \mathbf{m} and covariance matrix \mathbf{Q} , we denote the pdf by $\mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{Q})$, and for its circular complex Gaussian counterpart, we use $\mathcal{CN}(\mathbf{x}; \mathbf{m}, \mathbf{Q})$. Finally, we denote the expectation operation by $\mathbb{E}\{\cdot\}$, the Dirac delta by $\delta(\cdot)$, the real field by \mathbb{R} , and the complex field by \mathbb{C} .

II. GAUSSIAN-MIXTURE GAMP

We first introduce Gaussian-mixture (GM) GAMP, a key component of our overall approach, where the coefficients in $\mathbf{x} = [x_1, \dots, x_N]^\top$ are assumed to be i.i.d with marginal pdf

$$p_X(x; \lambda, \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}) = (1 - \lambda)\delta(x) + \lambda \sum_{\ell=1}^L \omega_\ell \mathcal{N}(x; \theta_\ell, \phi_\ell), \quad (3)$$

where $\delta(\cdot)$ is the Dirac delta, λ is the sparsity rate, and, for the k^{th} GM component, ω_k , θ_k , and ϕ_k are the weight, mean, and variance, respectively. In the sequel, we use $\boldsymbol{\omega} \triangleq [\omega_1, \dots, \omega_L]^\top$ and similar definitions for $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$. The noise $\mathbf{w} = [w_1, \dots, w_M]^\top$ is then assumed to be i.i.d Gaussian, with mean zero and variance ψ , i.e.,

$$p_W(w; \psi) = \mathcal{N}(w; 0, \psi), \quad (4)$$

and independent of \mathbf{x} . Although above and in the sequel we assume real-valued quantities, all expressions in the sequel can be converted to the circular-complex case by replacing \mathcal{N} with \mathcal{CN} and removing all $\frac{1}{2}$'s. We note that, from the perspective of GM-GAMP, the prior parameters $\mathbf{q} \triangleq [\lambda, \boldsymbol{\omega}, \boldsymbol{\theta}, \boldsymbol{\phi}, \psi]$ and the number of mixture components, L , are treated as fixed and known.

GAMP models the relationship between the m^{th} observed output y_m and the corresponding noiseless output $z_m \triangleq \mathbf{a}_m^\top \mathbf{x}$, where \mathbf{a}_m^\top denotes the m^{th} row of \mathbf{A} , using an arbitrary conditional pdf $p_{Y|Z}(y_m|z_m)$. Under the AWGN assumption¹ (4), $p_{Y|Z}(y|z) = \mathcal{N}(y; z, \psi)$, and thus the quantities $g_{\text{out}}(\cdot)$ and $g'_{\text{out}}(\cdot)$ in Table I become [14]

$$g_{\text{out}}(y, \hat{z}, \mu^z; \mathbf{q}) = \frac{y - \hat{z}}{\mu^z + \psi}, \quad g'_{\text{out}}(y, \hat{z}, \mu^z; \mathbf{q}) = \frac{-1}{\mu^z + \psi}. \quad (5)$$

As for the quantities $g_{\text{in}}(\cdot)$ and $g'_{\text{in}}(\cdot)$ in Table I, straightforward calculations with our GM signal prior (3) reveal that

$$g_{\text{in}}(\hat{r}, \mu^r; \mathbf{q}) = \pi(\hat{r}, \mu^r; \mathbf{q}) \frac{\sum_{\ell=1}^L \beta_\ell(\hat{r}, \mu^r; \mathbf{q}) \gamma_\ell(\hat{r}, \mu^r; \mathbf{q})}{\sum_{\ell=1}^L \beta_\ell(\hat{r}, \mu^r; \mathbf{q})} \quad (6)$$

$$\begin{aligned} \mu^r g'_{\text{in}}(\hat{r}, \mu^r; \mathbf{q}) &= -|g_{\text{in}}(\hat{r}, \mu^r; \mathbf{q})|^2 + \pi(\hat{r}, \mu^r; \mathbf{q}) \\ &\times \frac{\sum_{\ell=1}^L \beta_\ell(\hat{r}, \mu^r; \mathbf{q}) (|\gamma_\ell(\hat{r}, \mu^r; \mathbf{q})|^2 + \nu_\ell(\hat{r}, \mu^r; \mathbf{q}))}{\sum_{\ell=1}^L \beta_\ell(\hat{r}, \mu^r; \mathbf{q})} \end{aligned} \quad (7)$$

where

$$\beta_\ell(\hat{r}, \mu^r; \mathbf{q}) \triangleq \lambda \omega_\ell \mathcal{N}(\hat{r}; \theta_\ell, \phi_\ell + \mu^r) \quad (8)$$

$$\pi(\hat{r}, \mu^r; \mathbf{q}) \triangleq \frac{1}{1 + \left(\frac{\sum_{\ell=1}^L \beta_\ell(\hat{r}, \mu^r; \mathbf{q})}{(1-\lambda)\mathcal{N}(0; \hat{r}, \mu^r)} \right)^{-1}} \quad (9)$$

$$\gamma_\ell(\hat{r}, \mu^r; \mathbf{q}) \triangleq \frac{\hat{r}/\mu^r + \theta_\ell/\phi_\ell}{1/\mu^r + 1/\phi_\ell} \quad (10)$$

$$\nu_\ell(\hat{r}, \mu^r; \mathbf{q}) \triangleq \frac{1}{1/\mu^r + 1/\phi_\ell}. \quad (11)$$

Table I then implies that GM-GAMP's marginal posteriors are

$$p(x_n | \mathbf{y}; \mathbf{q}) = \frac{p_X(x_n; \mathbf{q}) \mathcal{N}(x_n; \hat{r}_n, \mu_n^r)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q})} \quad (12)$$

$$= \left((1 - \lambda)\delta(x_n) + \lambda \sum_{\ell=1}^L \omega_\ell \mathcal{N}(x_n; \theta_\ell, \phi_\ell) \right) \frac{\mathcal{N}(x_n; \hat{r}_n, \mu_n^r)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q})} \quad (13)$$

$$\zeta(\hat{r}, \mu^r; \mathbf{q}) \triangleq \int_x p_X(x; \mathbf{q}) \mathcal{N}(x; \hat{r}, \mu^r) \quad (14)$$

$$= (1 - \lambda)\mathcal{N}(0; \hat{r}, \mu^r) + \lambda \sum_{\ell=1}^L \omega_\ell \mathcal{N}(0; \hat{r} - \theta_\ell, \mu^r + \phi_\ell). \quad (15)$$

From (13), it is straightforward to show that the posterior support probabilities returned by GM-GAMP are

$$\Pr\{x_n \neq 0 | \mathbf{y}; \mathbf{q}\} = \pi(\hat{r}_n, \mu_n^r; \mathbf{q}). \quad (16)$$

In principle, one could specify GAMP for an arbitrary signal prior $p_X(\cdot)$. However, if the integrals in (D4)–(D6) are not computable in closed form (e.g., when $p_X(\cdot)$ is Student's-t), then they would need to be computed numerically, thereby drastically increasing the computational complexity of GAMP. In contrast, for GM signal models, we see from the above that all steps can be computed in closed form. Thus, a practical

¹Because GAMP can handle an arbitrary $p_{Y|Z}(\cdot|\cdot)$, the extension of EM-GM-AMP to additive non-Gaussian noise, and even non-additive measurement channels (such as with quantized outputs or logistic regression), is straightforward. Moreover, the parameters of the pdf $p_{Y|Z}(\cdot|\cdot)$ could be learned using a method similar to that which we propose for learning the AWGN variance ψ in Section III. Finally, one could even model $p_{Y|Z}(\cdot|\cdot)$ as a Gaussian mixture and learn the corresponding parameters.

definitions:	
$p_{Y Z}(z y; \hat{z}, \mu^z) = \frac{p_Y Z(y z) \mathcal{N}(z; \hat{z}, \mu^z)}{\int_{z'} p_Y Z(y z') \mathcal{N}(z'; \hat{z}, \mu^z)}$	(D1)
$g_{\text{out}}(y, \hat{z}, \mu^z) = \frac{1}{\mu^z} (\mathbb{E}_{Z Y} \{z y; \hat{z}, \mu^z\} - \hat{z})$	(D2)
$g'_{\text{out}}(y, \hat{z}, \mu^z) = \frac{1}{\mu^z} \left(\frac{\text{var}_{Z Y} \{z y; \hat{z}, \mu^z\}}{\mu^z} - 1 \right)$	(D3)
$p_{X Y}(x y; \hat{r}, \mu^r) = \frac{p_X(x) \mathcal{N}(x; \hat{r}, \mu^r)}{\int_{x'} p_X(x') \mathcal{N}(x'; \hat{r}, \mu^r)}$	(D4)
$g_{\text{in}}(\hat{r}, \mu^r) = \int_{\mathbf{x}} x p_X(\mathbf{x} y; \hat{r}, \mu^r)$	(D5)
$g'_{\text{in}}(\hat{r}, \mu^r) = \frac{1}{\mu^r} \int_{\mathbf{x}} x - g_{\text{in}}(\hat{r}, \mu^r) ^2 p_X(\mathbf{x} y; \hat{r}, \mu^r)$	(D6)
initialize:	
$\forall n : \hat{x}_n(1) = \int_{\mathbf{x}} x p_X(\mathbf{x})$	(I1)
$\forall n : \mu_n^x(1) = \int_{\mathbf{x}} x - \hat{x}_n(1) ^2 p_X(\mathbf{x})$	(I2)
$\forall m : \hat{u}_m(0) = 0$	(I3)
for $t = 1 : T_{\text{max}}$,	
$\forall m : \hat{z}_m(t) = \sum_{n=1}^N A_{mn} \hat{x}_n(t)$	(R1)
$\forall m : \mu_m^z(t) = \sum_{n=1}^N A_{mn} ^2 \mu_n^x(t)$	(R2)
$\forall m : \hat{p}_m(t) = \hat{z}_m(t) - \mu_m^z(t) \hat{u}_m(t-1)$	(R3)
$\forall m : \hat{u}_m(t) = g_{\text{out}}(y_m, \hat{p}_m(t), \mu_m^z(t))$	(R4)
$\forall m : \mu_m^u(t) = -g'_{\text{out}}(y_m, \hat{p}_m(t), \mu_m^z(t))$	(R5)
$\forall n : \mu_n^r(t) = \left(\sum_{m=1}^N A_{mn} ^2 \mu_m^u(t) \right)^{-1}$	(R6)
$\forall n : \hat{r}_n(t) = \hat{x}_n(t) + \mu_n^r(t) \sum_{m=1}^M A_{mn}^* \hat{u}_m(t)$	(R7)
$\forall n : \mu_n^x(t+1) = \mu_n^r(t) g'_{\text{in}}(\hat{r}_n(t), \mu_n^r(t))$	(R8)
$\forall n : \hat{x}_n(t+1) = g_{\text{in}}(\hat{r}_n(t), \mu_n^r(t))$	(R9)
if $\sum_{n=1}^N \hat{x}_n(t+1) - \hat{x}_n(t) ^2 < \tau_{\text{gamp}} \sum_{n=1}^N \hat{x}_n(t) ^2$, break	(R10)
end	

TABLE I
THE GAMP ALGORITHM [14]

approach to the use of GAMP with an intractable signal prior $p_X(\cdot)$ is to approximate $p_X(\cdot)$ using an L -term GM, after which all GAMP steps can be easily implemented. The same approach could also be used to ease the implementation of intractable noise priors $p_{Y|Z}(\cdot|\cdot)$.

III. EM LEARNING OF THE PRIOR PARAMETERS \mathbf{q}

In our approach, the expectation-maximization (EM) algorithm [15] is employed to learn the prior parameters $\mathbf{q} \triangleq [\lambda, \omega, \theta, \phi, \psi]$. The EM algorithm is an iterative technique that increases a lower bound on the likelihood $p(\mathbf{y}; \mathbf{q})$ at each iteration, thus guaranteeing that the likelihood converges to a local maximum. In our case, the “hidden data” is chosen as $\{\mathbf{x}, \mathbf{w}\}$, implying the iteration- i EM update

$$\mathbf{q}^{i+1} = \arg \max_{\mathbf{q}} \mathbb{E} \{ \ln p(\mathbf{x}, \mathbf{w}; \mathbf{q}) \mid \mathbf{y}; \mathbf{q}^i \}, \quad (17)$$

where $\mathbb{E}\{\cdot | \mathbf{y}; \mathbf{q}^i\}$ denotes expectation conditioned on the observations \mathbf{y} under the parameter hypothesis \mathbf{q}^i . Since it is impractical to update the entire vector \mathbf{q} at once, we update \mathbf{q} one component at a time (while holding the others fixed), which can be recognized as the “incremental” technique from [20]. In the sequel, we use “ $\mathbf{q}_{\setminus \lambda}^i$ ” to denote the vector \mathbf{q}^i with the element λ removed (and similar for the other parameters). We emphasize that our EM-guided GM-fitting procedure differs from the standard one (e.g., [21, p. 435]) because, in our case, realizations of \mathbf{x} are not directly observed.

A. EM Update of the Gaussian Noise Variance ψ

We first derive the EM update for the noise variance ψ given a previous parameter estimate \mathbf{q}^i . Because \mathbf{w} is apriori independent of \mathbf{x} and i.i.d, we can write $p(\mathbf{x}, \mathbf{w}; \mathbf{q}) = C \prod_{m=1}^M p_W(w_m; \psi)$ for a ψ -invariant constant C , and so

$$\psi^{i+1} = \arg \max_{\psi > 0} \sum_{m=1}^M \mathbb{E} \{ \ln p_W(w_m; \psi) \mid \mathbf{y}; \mathbf{q}^i \}. \quad (18)$$

The maximizing value of ψ in (18) is necessarily a value of ψ that zeroes the derivative of the sum, i.e., that satisfies

$$\sum_{m=1}^M \int_{w_m} p(w_m | \mathbf{y}; \mathbf{q}^i) \frac{d}{d\psi} \ln p_W(w_m; \psi) = 0. \quad (19)$$

Because $p_W(w_m; \psi) = \mathcal{N}(w_m; 0, \psi)$, it is readily seen that

$$\frac{d}{d\psi} \ln p_W(w_m; \psi) = \frac{1}{2} \left(\frac{|w_m|^2}{\psi^2} - \frac{1}{\psi} \right), \quad (20)$$

which, when plugged into (19), yields the unique solution

$$\psi^{i+1} = \frac{1}{M} \sum_{m=1}^M \int_{w_m} |w_m|^2 p(w_m | \mathbf{y}; \mathbf{q}^i). \quad (21)$$

Since $w_m = y_m - z_m$ for $z_m \triangleq \mathbf{a}_m^T \mathbf{x}$, we can also write

$$\begin{aligned} \psi^{i+1} &= \frac{1}{M} \sum_{m=1}^M \int_{z_m} |y_m - z_m|^2 p(z_m | \mathbf{y}; \mathbf{q}^i) \\ &= \frac{1}{M} \sum_{m=1}^M (|y_m - \hat{z}_m|^2 + \mu_m^z) \end{aligned} \quad (22)$$

where \hat{z}_m and μ_m^z , the posterior mean and variance of z_m , are available from GAMP (see steps (R1)-(R2) in Table I).

B. EM Updates of the Signal Parameters: BG Case

Suppose that the signal distribution $p_X(\cdot)$ is modeled using an $L = 1$ -term GM, i.e., a Bernoulli-Gaussian (BG) pdf.² In this case, the marginal signal prior in (3) reduces to

$$p_X(x; \lambda, \omega, \theta, \phi) = (1 - \lambda) \delta(x) + \lambda \mathcal{N}(x; \theta, \phi). \quad (24)$$

Note that, in the BG case, the mixture weight ω is, by definition, unity and does not need to be learned.

We now derive the EM update for λ given previous parameters $\mathbf{q}^i \triangleq [\lambda^i, \theta^i, \phi^i, \psi^i]$. Because \mathbf{x} is apriori independent of \mathbf{w} and i.i.d, we can write $p(\mathbf{x}, \mathbf{w}; \mathbf{q}) = C \prod_{n=1}^N p_X(x_n; \lambda, \theta, \phi)$ for a λ -invariant constant C , and so

$$\lambda^{i+1} = \arg \max_{\lambda \in (0,1)} \sum_{n=1}^N \mathbb{E} \{ \ln p_X(x_n; \lambda, \mathbf{q}_{\setminus \lambda}^i) \mid \mathbf{y}; \mathbf{q}^i \}. \quad (25)$$

The maximizing value of λ in (25) is necessarily a value of λ that zeroes the derivative of the sum, i.e., that satisfies

$$\sum_{n=1}^N \int_{x_n} p(x_n | \mathbf{y}; \mathbf{q}^i) \frac{d}{d\lambda} \ln p_X(x_n; \lambda, \mathbf{q}_{\setminus \lambda}^i) = 0. \quad (26)$$

For the BG $p_X(x_n; \lambda, \theta, \phi)$ in (24), it is readily seen that

$$\frac{d}{d\lambda} \ln p_X(x_n; \lambda, \mathbf{q}_{\setminus \lambda}^i) = \frac{\mathcal{N}(x_n; \theta^i, \phi^i) - \delta(x_n)}{p_X(x_n; \lambda, \mathbf{q}_{\setminus \lambda}^i)} = \begin{cases} \frac{1}{1-\lambda} & x_n \neq 0 \\ \frac{-1}{1-\lambda} & x_n = 0. \end{cases} \quad (27)$$

Plugging (27) and (13) into (26), it becomes evident that the neighborhood around the point $x_n = 0$ should be treated differently than the remainder of \mathbb{R} . Thus, we define the closed

²We note that, after presenting our initial work on EM-BG-AMP in [1] and submitting it to [2], a closely related approach appeared in [22] in the context of “seeded belief propagation.” Although the main contribution of our current work is EM-GM-AMP, we detail the case of EM-BG-AMP in order to demonstrate that the approximations needed for the EM-GM recursion become tight in the $L = 1$ case (i.e., EM-BG).

ball $\mathcal{B}_\epsilon \triangleq [-\epsilon, \epsilon]$ and its complement $\overline{\mathcal{B}}_\epsilon \triangleq \mathbb{R} \setminus \mathcal{B}_\epsilon$, and note that, in the limit $\epsilon \rightarrow 0$, the following is equivalent to (26):

$$\frac{1}{\lambda} \sum_{n=1}^N \underbrace{\int_{x_n \in \mathcal{B}_\epsilon} p(x_n | \mathbf{y}; \mathbf{q}^i)}_{\stackrel{\epsilon \rightarrow 0}{=} \pi(\hat{r}_n, \mu_n^r; \mathbf{q}^i)} = \frac{1}{1-\lambda} \sum_{n=1}^N \underbrace{\int_{x_n \in \overline{\mathcal{B}}_\epsilon} p(x_n | \mathbf{y}; \mathbf{q}^i)}_{\stackrel{\epsilon \rightarrow 0}{=} 1 - \pi(\hat{r}_n, \mu_n^r; \mathbf{q}^i)}. \quad (28)$$

To verify that the left integral converges to the $\pi(\hat{r}_n, \mu_n^r; \mathbf{q}^i)$ defined in (9), it suffices to plug (13) into (28) and apply the Gaussian-pdf multiplication rule.³ The right integral must then equal one minus the value of the left integral because, together, their regions of integration cover the real line. Finally, the EM update for λ is the unique value satisfying (28) as $\epsilon \rightarrow 0$, which is readily shown to be

$$\lambda^{i+1} = \frac{1}{N} \sum_{n=1}^N \pi(\hat{r}_n, \mu_n^r; \mathbf{q}^i). \quad (29)$$

Conveniently, the quantities $\{\pi(\hat{r}_n, \mu_n^r; \mathbf{q}^i)\}_{n=1}^N$ are GM-GAMP outputs, as per (16) and (9).

Similar to (25), the EM update for θ can be written as

$$\theta^{i+1} = \arg \max_{\theta \in \mathbb{R}} \sum_{n=1}^N \mathbb{E} \{ \ln p_X(x_n; \theta, \mathbf{q}_\theta^i) | \mathbf{y}; \mathbf{q}^i \}. \quad (30)$$

The maximizing value of θ in (30) is again a necessarily a value of θ that zeroes the derivative, i.e., that satisfies

$$\sum_{n=1}^N \int_{x_n} p(x_n | \mathbf{y}; \mathbf{q}^i) \frac{d}{d\theta} \ln p_X(x_n; \theta, \mathbf{q}_\theta^i) = 0. \quad (31)$$

For the BG $p_X(x_n; \lambda, \theta, \phi)$ given in (24),

$$\frac{d}{d\theta} \ln p_X(x_n; \lambda^i, \theta, \phi^i) = \frac{(x_n - \theta)}{\phi^i} \frac{\lambda^i \mathcal{N}(x_n; \theta, \phi^i)}{p_X(x_n; \theta, \mathbf{q}_\theta^i)} = \begin{cases} \frac{x_n - \theta}{\phi^i} & x_n \neq 0 \\ 0 & x_n = 0. \end{cases} \quad (32)$$

Splitting the domain of integration in (31) into \mathcal{B}_ϵ and $\overline{\mathcal{B}}_\epsilon$ as before, and then plugging in (32), we find that the following is equivalent to (31) in the limit of $\epsilon \rightarrow 0$:

$$\sum_{n=1}^N \int_{x_n \in \overline{\mathcal{B}}_\epsilon} (x_n - \theta) p(x_n | \mathbf{y}; \mathbf{q}^i) = 0. \quad (33)$$

The unique value of θ satisfying (33) as $\epsilon \rightarrow 0$ is then

$$\theta^{i+1} = \frac{\sum_{n=1}^N \lim_{\epsilon \rightarrow 0} \int_{x_n \in \overline{\mathcal{B}}_\epsilon} x_n p(x_n | \mathbf{y}; \mathbf{q}^i)}{\sum_{n=1}^N \lim_{\epsilon \rightarrow 0} \int_{x_n \in \overline{\mathcal{B}}_\epsilon} p(x_n | \mathbf{y}; \mathbf{q}^i)} \quad (34)$$

$$= \frac{1}{\lambda^{i+1} N} \sum_{n=1}^N \pi(\hat{r}_n, \mu_n^r; \mathbf{q}^i) \gamma_1(\hat{r}_n, \mu_n^r; \mathbf{q}^i) \quad (35)$$

for the GM-GAMP outputs $\{\gamma_1(\hat{r}_n, \mu_n^r; \mathbf{q}^i)\}_{n=1}^N$ defined in (10). The equality in (35) can be verified by plugging the GAMP posterior expression from (13) into (34) and simplifying via the Gaussian-pdf multiplication rule.

Similar to (25), the EM update for ϕ can be written as

$$\hat{\phi}^{i+1} = \arg \max_{\phi > 0} \sum_{n=1}^N \mathbb{E} \{ \ln p_X(x_n; \phi, \mathbf{q}_\phi^i) | \mathbf{y}; \mathbf{q}^i \}. \quad (36)$$

The maximizing value of ϕ in (36) is again necessarily a value of ϕ that zeroes the derivative, i.e., that satisfies

$$\sum_{n=1}^N \int_{x_n} p(x_n | \mathbf{y}; \mathbf{q}^i) \frac{d}{d\phi} \ln p_X(x_n; \phi, \mathbf{q}_\phi^i) = 0. \quad (37)$$

For the $p_X(x_n; \lambda, \theta, \phi)$ given in (24), it is readily seen that

$$\begin{aligned} \frac{d}{d\phi} \ln p_X(x_n; \lambda^i, \theta^i, \phi) &= \frac{1}{2} \left(\frac{|x_n - \theta^i|^2}{(\phi)^2} - \frac{1}{\phi} \right) \frac{\lambda^i \mathcal{N}(x_n; \theta^i, \phi)}{p_X(x_n; \phi, \mathbf{q}_\phi^i)} \\ &= \begin{cases} \frac{1}{2} \left(\frac{|x_n - \theta^i|^2}{(\phi)^2} - \frac{1}{\phi} \right) & x_n \neq 0 \\ 0 & x_n = 0 \end{cases}. \end{aligned} \quad (38)$$

Splitting the domain of integration in (37) into \mathcal{B}_ϵ and $\overline{\mathcal{B}}_\epsilon$ as before, and then plugging in (38), we find that the following is equivalent to (37) in the limit of $\epsilon \rightarrow 0$:

$$\sum_{n=1}^N \int_{x_n \in \overline{\mathcal{B}}_\epsilon} (|x_n - \theta^i|^2 - \phi) p(x_n | \mathbf{y}; \mathbf{q}^i) = 0. \quad (39)$$

The unique value of ϕ satisfying (39) as $\epsilon \rightarrow 0$ is then

$$\phi^{i+1} = \frac{\sum_{n=1}^N \lim_{\epsilon \rightarrow 0} \int_{x_n \in \overline{\mathcal{B}}_\epsilon} |x_n - \theta^i|^2 p(x_n | \mathbf{y}; \mathbf{q}^i)}{\sum_{n=1}^N \lim_{\epsilon \rightarrow 0} \int_{x_n \in \overline{\mathcal{B}}_\epsilon} p(x_n | \mathbf{y}; \mathbf{q}^i)} \quad (40)$$

Finally, we expand $|x_n - \theta^i|^2 = |x_n|^2 - 2 \operatorname{Re}(x_n^* \theta^i) + |\theta^i|^2$ which gives

$$\phi^{i+1} = \frac{1}{\lambda^{i+1} N} \sum_{n=1}^N \pi(\hat{r}_n, \mu_n^r; \mathbf{q}^i) (|\theta^i - \gamma_1(\hat{r}_n, \mu_n^r; \mathbf{q}^i)|^2 + \nu_1(\hat{r}_n, \mu_n^r; \mathbf{q}^i)) \quad (41)$$

for the GM-GAMP outputs $\{\nu_1(\hat{r}_n, \mu_n^r; \mathbf{q}^i)\}_{n=1}^N$ defined in (11). The equality in (41) can be verified by plugging the GAMP posterior expression from (13) into (40) and simplifying using the Gaussian-pdf multiplication rule.

C. EM Updates of the Signal Parameters: GM Case

We now generalize the EM updates derived in Section III-B to the GM prior given in (3) for $L \geq 1$. As we shall see, it is not possible to write the exact EM updates in closed-form when $L > 1$, and so some approximations will be made.

We begin by deriving the EM update for λ given the previous parameters $\mathbf{q}^i \triangleq [\lambda^i, \omega^i, \theta^i, \phi^i, \psi^i]$. The first two steps are identical to the steps (25) and (26) presented for the BG case, and for brevity we do not repeat them here. In the third step, use of the GM prior (3) yields

$$\frac{d}{d\lambda} \ln p_X(x_n; \lambda, \mathbf{q}_\lambda^i) = \frac{\sum_{\ell=1}^L \omega_\ell^i \mathcal{N}(x_n; \theta_\ell^i, \phi_\ell^i) - \delta(x_n)}{p_X(x_n; \lambda, \mathbf{q}_\lambda^i)} = \begin{cases} \frac{1}{\lambda} & x_n \neq 0 \\ \frac{-1}{1-\lambda} & x_n = 0 \end{cases}, \quad (42)$$

which coincides with the corresponding BG expression (27). The remaining steps also coincide with those in the BG case, and so the final EM update for λ , in the case of a GM,⁴ is given by (29).

We next derive the EM updates for the Gaussian-Mixture parameters ω , θ , and ϕ . For each $k = 1, \dots, L$, we incrementally update θ_k , then ϕ_k , and then the entire vector ω , while

⁴The arguments in this section reveal that, under signal priors of the form $p_X(x) = (1-\lambda)\delta(x) + \lambda f_X(x)$, where $f_X(\cdot)$ can be arbitrary, the EM update for λ is that given in (29).

³ $\mathcal{N}(x; a, A) \mathcal{N}(x; b, B) = \mathcal{N}(x; \frac{a/A + b/B}{1/A + 1/B}, \frac{1}{1/A + 1/B}) \mathcal{N}(0; a-b, A+B)$.

holding all other parameters fixed. The EM updates are thus

$$\theta_k^{i+1} = \arg \max_{\theta_k \in \mathbb{R}} \sum_{n=1}^N \mathbb{E} \{ \ln p_X(x_n; \theta_k, \mathbf{q}_{\setminus \theta_k}^i | \mathbf{y}; \mathbf{q}^i) \}, \quad (43)$$

$$\phi_k^{i+1} = \arg \max_{\phi_k > 0} \sum_{n=1}^N \mathbb{E} \{ \ln p_X(x_n; \phi_k, \mathbf{q}_{\setminus \phi_k}^i | \mathbf{y}; \mathbf{q}^i) \} \quad (44)$$

$$\omega^{i+1} = \arg \max_{\omega > 0: \sum_k \omega_k = 1} \sum_{n=1}^N \mathbb{E} \{ \ln p_X(x_n; \omega, \mathbf{q}_{\setminus \omega}^i | \mathbf{y}; \mathbf{q}^i) \} \quad (45)$$

Following (31), the maximizing value of θ_k in (43) is again necessarily a value of θ_k that zeroes the derivative, i.e.,

$$\sum_{n=1}^N \int_{x_n} p(x_n | \mathbf{y}; \mathbf{q}^i) \frac{d}{d\theta_k} \ln p_X(x_n; \theta_k, \mathbf{q}_{\setminus \theta_k}^i) = 0, \quad (46)$$

where $p(x_n | \mathbf{y}; \mathbf{q}^i) = p_X(x_n; \mathbf{q}^i) \mathcal{N}(x_n; \hat{r}_n, \mu_n^r) / \zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i)$ from (D4), with GM $p_X(x; \mathbf{q})$ from (3) and $\zeta(\hat{r}, \mu^r; \mathbf{q})$ from (15). Taking the derivative, we find

$$\begin{aligned} \frac{d}{d\theta_k} \ln p_X(x_n; \theta_k, \mathbf{q}_{\setminus \theta_k}^i) &= \left(\frac{x_n - \theta_k}{\phi_k^i} \right) \\ &\times \frac{\lambda^i \omega_k^i \mathcal{N}(x_n; \theta_k, \phi_k^i)}{(1 - \lambda^i) \delta(x_n) + \lambda^i (\omega_k^i \mathcal{N}(x_n; \theta_k, \phi_k^i) + \sum_{\ell \neq k} \omega_\ell^i \mathcal{N}(x_n; \theta_\ell^i, \phi_\ell^i))}. \end{aligned} \quad (47)$$

Integrating (46) separately over \mathcal{B}_ϵ and $\overline{\mathcal{B}_\epsilon}$, as in (28), and taking $\epsilon \rightarrow 0$, we find that the \mathcal{B}_ϵ portion vanishes, giving the necessary condition

$$\sum_{n=1}^N \int_{x_n} \frac{p(x_n | x_n \neq 0, \mathbf{y}; \mathbf{q}^i) \lambda^i \omega_k^i \mathcal{N}(x_n; \theta_k, \phi_k^i) (x_n - \theta_k)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i) (\omega_k^i \mathcal{N}(x_n; \theta_k, \phi_k^i) + \sum_{\ell \neq k} \omega_\ell^i \mathcal{N}(x_n; \theta_\ell^i, \phi_\ell^i))} = 0. \quad (48)$$

Since this integral cannot be evaluated in closed form, we apply the approximation $\mathcal{N}(x_n; \theta_k, \phi_k^i) \approx \mathcal{N}(x_n; \theta_k^i, \phi_k^i)$ in both the numerator and denominator, and subsequently exploit the fact that $p(x_n | x_n \neq 0, \mathbf{y}; \mathbf{q}^i) = \mathcal{N}(x_n; \hat{r}_n, \mu_n^r) \sum_{\ell} \omega_\ell^i \mathcal{N}(x_n; \theta_\ell^i, \phi_\ell^i)$ to cancel terms, giving the (approximated) necessary condition

$$\sum_{n=1}^N \int_{x_n} \frac{\lambda^i \omega_k^i \mathcal{N}(x_n; \hat{r}_n, \mu_n^r) \mathcal{N}(x_n; \theta_k^i, \phi_k^i)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i)} (x_n - \theta_k) = 0. \quad (49)$$

We then simplify (49) using the Gaussian-pdf multiplication rule, and set θ_k^{i+1} equal to the value of θ_k that satisfies (49):

$$\theta_k^{i+1} = \frac{\sum_{n=1}^N \Pr\{x_n \neq 0, k_n = k | \mathbf{y}; \mathbf{q}^i\} \gamma_k(\hat{r}_n, \mu_n^r; \mathbf{q}^i)}{\sum_{n=1}^N \Pr\{x_n \neq 0, k_n = k | \mathbf{y}; \mathbf{q}^i\}}, \quad (50)$$

where we define

$$\Pr\{x_n \neq 0, k_n = k | \mathbf{y}; \mathbf{q}^i\} \triangleq \pi(\hat{r}_n, \mu_n^r; \mathbf{q}^i) \frac{\beta_k(\hat{r}_n, \mu_n^r; \mathbf{q}^i)}{\sum_{\ell=1}^L \beta_\ell(\hat{r}_n, \mu_n^r; \mathbf{q}^i)} \quad (51)$$

using $\beta_k(\hat{r}, \mu^r; \mathbf{q}^i)$ from (8) and $\gamma_k(\hat{r}, \mu^r; \mathbf{q}^i)$ from (10). Here, the notation “ $k_n = k$ ” describes the event that x_n was generated from mixture component k .

For sparse signals \mathbf{x} , we find that learning the GM means $\{\theta_k\}$ using the above EM procedure yields excellent recovery MSE. However, for “heavy-tailed” signals, such as those generated from Student’s t-distributions, our experience indicates that the EM-learned values of $\{\theta_k\}$ tend to gravitate towards the outliers in $\{x_n\}_{n=1}^N$, resulting in an overfitting of $p_X(\cdot)$ and thus poor reconstruction MSE. For such heavy-tailed signals,

we find that better reconstruction performance is obtained by fixing the means at zero (i.e., $\theta_k^i = 0 \ \forall k, i$). Thus, in the remainder of the paper, we consider two modes of operation: a “sparse” mode where θ is learned via the above EM procedure, and a “heavy-tailed” mode that fixes $\theta = \mathbf{0}$.

Following (46), the maximizing value of ϕ_k in (44) is necessarily a value of ϕ_k that zeroes the derivative, i.e.,

$$\sum_{n=1}^N \int_{x_n} p(x_n | \mathbf{y}; \mathbf{q}^i) \frac{d}{d\phi_k} \ln p_X(x_n; \phi_k, \mathbf{q}_{\setminus \phi_k}^i) = 0. \quad (52)$$

As for the derivative in the previous expression, we find

$$\begin{aligned} \frac{d}{d\phi_k} \ln p_X(x_n; \phi_k, \mathbf{q}_{\setminus \phi_k}^i) &= \frac{1}{2} \left(\frac{|x_n - \theta_k^i|^2}{\phi_k^2} - \frac{1}{\phi_k} \right) \\ &\times \frac{\lambda^i \omega_k^i \mathcal{N}(x_n; \theta_k^i, \phi_k)}{(1 - \lambda^i) \delta(x_n) + \lambda^i (\omega_k^i \mathcal{N}(x_n; \theta_k^i, \phi_k) + \sum_{\ell \neq k} \omega_\ell^i \mathcal{N}(x_n; \theta_\ell^i, \phi_\ell^i))}. \end{aligned} \quad (53)$$

Integrating (52) separately over \mathcal{B}_ϵ and $\overline{\mathcal{B}_\epsilon}$, as in (28), and taking $\epsilon \rightarrow 0$, we find that the \mathcal{B}_ϵ portion vanishes, giving

$$\sum_{n=1}^N \int_{x_n} \frac{p(x_n | x_n \neq 0, \mathbf{y}; \mathbf{q}^i) \lambda^i \omega_k^i \mathcal{N}(x_n; \theta_k^i, \phi_k)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i) (\omega_k^i \mathcal{N}(x_n; \theta_k^i, \phi_k) + \sum_{\ell \neq k} \omega_\ell^i \mathcal{N}(x_n; \theta_\ell^i, \phi_\ell^i))} \left(\frac{|x_n - \theta_k^i|^2}{\phi_k} - 1 \right) = 0. \quad (54)$$

Similar to (48), this integral is difficult to evaluate, and so we again apply the approximation $\mathcal{N}(x_n; \theta_k^i, \phi_k) \approx \mathcal{N}(x_n; \theta_k^i, \phi_k^i)$ in the numerator and denominator, after which several terms cancel, yielding the necessary condition

$$\sum_{n=1}^N \int_{x_n} \frac{\mathcal{N}(x_n; \hat{r}_n, \mu_n^r) \lambda^i \omega_k^i \mathcal{N}(x_n; \theta_k^i, \phi_k^i)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i)} \left(\frac{|x_n - \theta_k^i|^2}{\phi_k} - 1 \right) = 0. \quad (55)$$

To find the value of ϕ_k satisfying (55), we expand $|x_n - \theta_k^i|^2 = |x_n|^2 - 2 \operatorname{Re}(x_n^* \theta_k^i) + |\theta_k^i|^2$ and apply the Gaussian-pdf multiplication rule, which gives

$$\phi_k^{i+1} = \frac{\sum_{n=1}^N \Pr\{x_n \neq 0, k_n = k | \mathbf{y}; \mathbf{q}^i\} (|\theta_k^i - \gamma_k(\hat{r}_n, \mu_n^r; \mathbf{q}^i)|^2 + \nu_k(\hat{r}_n, \mu_n^r; \mathbf{q}^i))}{\sum_{n=1}^N \Pr\{x_n \neq 0, k_n = k | \mathbf{y}; \mathbf{q}^i\}} \quad (56)$$

for $\Pr\{x_n \neq 0, k_n = k | \mathbf{y}; \mathbf{q}^i\}$ defined in (51).

Finally, the value of the positive ω maximizing (45) under the pmf constraint $\sum_{k=1}^L \omega_k = 1$ can be found by solving the unconstrained optimization problem $\max_{\omega, \xi} J(\omega, \xi)$, where ξ is a Lagrange multiplier and

$$J(\omega, \xi) \triangleq \sum_{n=1}^N \mathbb{E} \{ \ln p_X(x_n; \omega, \mathbf{q}_{\setminus \omega}^i | \mathbf{y}; \mathbf{q}^i) \} - \xi \left(\sum_{\ell=1}^L \omega_\ell - 1 \right) \quad (57)$$

$$= \sum_{n=1}^N \int_{x_n} p(x_n | \mathbf{y}; \mathbf{q}^i) \ln p_X(x_n; \omega, \mathbf{q}_{\setminus \omega}^i) - \xi \left(\sum_{\ell=1}^L \omega_\ell - 1 \right). \quad (58)$$

We start by setting $\frac{d}{d\omega_k} J(\omega, \xi) = 0$, which yields

$$\sum_{n=1}^N \int_{x_n} \frac{p_X(x_n; \mathbf{q}^i) \mathcal{N}(x_n; \hat{r}_n, \mu_n^r)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i)} \frac{d}{d\omega_k} \ln p_X(x_n; \omega, \mathbf{q}_{\setminus \omega}^i) = \xi. \quad (59)$$

$$\Leftrightarrow \sum_{n=1}^N \int_{x_n} \frac{p_X(x_n; \mathbf{q}^i) \mathcal{N}(x_n; \hat{r}_n, \mu_n^r)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i)} \frac{\lambda^i \mathcal{N}(x_n; \theta_k^i, \phi_k^i)}{p_X(x_n; \omega, \mathbf{q}_{\setminus \omega}^i)} = \xi. \quad (60)$$

Like in (48) and (54), the above integral is difficult to evaluate, and so we approximate $\omega \approx \omega^i$, which reduces the previous equation to

$$\xi = \sum_{n=1}^N \int_{x_n} \frac{\lambda^i \mathcal{N}(x_n; \theta_k^i, \phi_k^i) \mathcal{N}(x_n; \hat{r}_n, \mu_n^r)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i)}. \quad (61)$$

Multiplying both sides by ω_k^i for $k = 1, \dots, L$, summing over k , employing the fact $1 = \sum_k \omega_k^i$, and simplifying, we obtain the equivalent condition

$$\xi = \sum_{n=1}^N \int_{x_n} \frac{\lambda^i \sum_{k=1}^L \omega_k^i \mathcal{N}(x_n; \theta_k^i, \phi_k^i) \mathcal{N}(x_n; \hat{r}_n, \mu_n^r)}{\zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i)} \quad (62)$$

$$= \sum_{n=1}^N \Pr\{x_n \neq 0 | \mathbf{y}; \mathbf{q}^i\}. \quad (63)$$

Plugging (63) into (61) and multiplying both sides by ω_k , the derivative-zeroing value of ω_k is seen to be

$$\omega_k = \frac{\sum_{n=1}^N \int_{x_n} \lambda^i \omega_k \mathcal{N}(x_n; \theta_k^i, \phi_k^i) \mathcal{N}(x_n; \hat{r}_n, \mu_n^r) / \zeta(\hat{r}_n, \mu_n^r; \mathbf{q}^i)}{\sum_{n=1}^N \Pr\{x_n \neq 0 | \mathbf{y}; \mathbf{q}^i\}}, \quad (64)$$

where, if we use $\omega_k \approx \omega_k^i$ on the right of (64), then we obtain

$$\omega_k^{i+1} = \frac{\sum_{n=1}^N \Pr\{x_n \neq 0, k_n = k | \mathbf{y}; \mathbf{q}^i\}}{\sum_{n=1}^N \Pr\{x_n \neq 0 | \mathbf{y}; \mathbf{q}^i\}}. \quad (65)$$

Note that the numerator and denominator of (65) can be computed from GM-GAMP outputs via (51), (16), and (9).

Although, for the case of GM priors, approximations were used in the derivation of the EM updates (50), (56), and (65), it is interesting to note that, in the case of $L = 1$ mixture components, these approximate EM-GM updates coincide with the *exact* EM-BG updates derived in Section III-B. In particular, the approximate-EM update of the GM parameter θ_1 in (50) coincides with the exact-EM update of the BG parameter θ in (35), the approximate-EM update of the GM parameter ϕ_1 in (56) coincides with the exact-EM update of the BG parameter ϕ in (41), and the approximate-EM update of the GM parameter ω_1 in (65) reduces to the fixed value 1. Thus, one can safely use the GM updates above in the BG setting without any loss of optimality.

D. EM Initialization

Since the EM algorithm is guaranteed to converge to only a local maximum of the likelihood function, proper initialization of the unknown parameters \mathbf{q} is essential. Here, we propose initialization strategies for both the “sparse” and “heavy-tailed” modes of operation, for a given value of L .

For the “sparse” mode, we set the initial sparsity rate λ^0 equal to the theoretical noiseless Lasso PTC, i.e., $\lambda^0 = \frac{M}{N} \rho_{SE}(\frac{M}{N})$, where [10]

$$\rho_{SE}(\frac{M}{N}) = \max_{c>0} \frac{1 - \frac{2N}{M} [(1+c^2)\Phi(c) - c\phi(c)]}{1 - c^2 - 2[(1+c^2)\Phi(c) - c\phi(c)]} \quad (66)$$

describes the maximum value of $\frac{K}{M}$ supported by Lasso for a given $\frac{M}{N}$, and where $\Phi(\cdot)$ and $\phi(\cdot)$ denote the cdf and pdf of the $\mathcal{N}(0, 1)$ distribution, respectively. Using the energies $\|\mathbf{y}\|_2^2$

and $\|\mathbf{A}\|_F^2$ and an assumed value of SNR^0 , we initialize the noise and signal variances, respectively, as

$$\psi^0 = \frac{\|\mathbf{y}\|_2^2}{(\text{SNR}^0 + 1)M}, \quad \varphi^0 = \frac{\|\mathbf{y}\|_2^2 - M\psi^0}{\|\mathbf{A}\|_F^2 \lambda^0}, \quad (67)$$

where, in the absence of (user provided) knowledge about the true $\text{SNR} \triangleq \|\mathbf{A}\mathbf{x}\|_2^2 / \|\mathbf{w}\|_2^2$, we suggest $\text{SNR}^0 = 100$. Then, we uniformly space the initial GM means θ^0 over $[-\frac{L+1}{2L}, \frac{L-1}{2L}]$, and subsequently fit the mixture weights ω^0 and variances ϕ^0 to the uniform pdf supported on $[-0.5, 0.5]$ (which can be done offline using the standard approach to EM-fitting of GM parameters, e.g., [21, p. 435]). Finally, we multiply θ^0 by $\sqrt{12\varphi^0}$ and ϕ^0 by $12\varphi^0$ to ensure that the resulting signal variance equals φ^0 .

For the “heavy-tailed” mode, we initialize λ^0 and ψ^0 as above and set, for $k = 1, \dots, L$,

$$\omega_k^0 = \frac{1}{L}, \quad \phi_k^0 = \frac{k}{\sqrt{L}} \frac{(\|\mathbf{y}\|_2^2 - M\psi^0)}{\|\mathbf{A}\|_F^2 \lambda^0}, \quad \text{and } \theta_k^0 = 0. \quad (68)$$

E. Selection of GM Model Order L

Until now, the GM model order L has been treated as fixed and known. Indeed, one approach to model-order selection is to choose a fixed value of L that is thought to be large enough to model the essential structure within the true signal pdf and, after an appropriate initialization of the $3L + 2$ parameters in \mathbf{q} (such as that described in Section III-D), to apply the EM-GM-AMP algorithm summarized in Table II.

An alternative approach to model-order selection is to start with the model order $L = 1$ (i.e., Bernoulli-Gaussian $p_X(\cdot)$) and increment L in steps of one, stopping as soon as negligible benefits are observed (e.g., $\|\hat{\mathbf{x}}_L - \hat{\mathbf{x}}_{L-1}\|_2^2 / \|\hat{\mathbf{x}}_{L-1}\|_2^2 < \tau_{01}$) or a predefined L_{\max} has been reached. Here, EM-GM-AMP would be re-run as described above for each new value of L . This “greedy” approach would relieve the user of the task of choosing an appropriate L and initializing the corresponding $3L + 2$ parameters in \mathbf{q} . In the remainder of this section, we propose a particular instantiation of this greedy approach.

When growing the model-order from L to $L + 1$, we choose to split the mixture component $k_* \in \{1, \dots, L\}$ with the “worst fit” into two new components. As a criterion for the worst-fitting mixture component, one could use local Kullback-Leibler divergence, as in Ueda’s split-and-merge-EM algorithm [23], or similar. Using k_* to denote the index of the mixture-component-to-split, the subset of signal coefficient indices n that are a-posteriori most-probably associated with k_* is identified, i.e.,

$$\mathfrak{N}_{k_*} \triangleq \{n \in \mathfrak{N} : \arg \max_k \Pr\{x_n \neq 0, k_n = k | \mathbf{y}; \mathbf{q}\} = k_*\}, \quad (69)$$

where $\mathfrak{N} \triangleq \{n : \Pr\{x_n \neq 0 | \mathbf{y}; \mathbf{q}\} > 0.5\}$ is the subset of coefficient indices that are a-posteriori most-probably non-zero (i.e., the GAMP support estimate) and where \mathbf{q} denotes the current estimate of the parameters. To simplify the notation, we henceforth assume, without loss of generality, that $k_* = L$.

To split the L^{th} mixture component, we then replace its mean θ_L , variance ϕ_L , and weight ω_L with two new values for each (i.e., θ_L^{new} and $\theta_{L+1}^{\text{new}}$ would replace θ_L), resulting in a new parameter vector \mathbf{q}^{new} containing $3(L+1)+2$ terms. Moreover,

rather than considering only a single possibility for \mathbf{q}^{new} , we allow that $S \geq 1$ hypotheses $\{\mathbf{q}_s^{\text{new}}\}_{s=1}^S$ are considered, where the values in each $\mathbf{q}_s^{\text{new}}$ are produced by some variation on the following two strategies:

- 1) Split-mean(a): $\theta_L^{\text{new}} = \theta_L - a$, $\theta_{L+1}^{\text{new}} = \theta_L + a$, $\phi_L^{\text{new}} = \phi_{L+1}^{\text{new}} = \phi_L$, and $\omega_{L+1}^{\text{new}} = \omega_L^{\text{new}} = \omega_L/2$;
- 2) Split-variance(b): $\phi_L^{\text{new}} = b\phi_L$ and $\phi_{L+1}^{\text{new}} = b^{-1}\phi_L$, $\theta_L^{\text{new}} = \theta_{L+1}^{\text{new}} = \theta_L$, and $\omega_L^{\text{new}} = \omega_{L+1}^{\text{new}} = \omega_L/2$,

where $a, b > 0$ are design parameters. Note that, by considering several distinct values of a and/or b , we get $S > 2$. Finally, to choose among the hypothesized splits $\{\mathbf{q}_s^{\text{new}}\}_{s=1}^S$, one could, e.g., select the one that maximizes the likelihood $E\{\ln p_X(\mathbf{x}; \mathbf{q}_s^{\text{new}}) | \mathbf{y}; \mathbf{q}\}$.

We investigated the performance of EM-GM-AMP under this “greedy” model-order selection procedure in [3]. Comparing the numerical results in [3] (where greedy model-order selection was used for the “sparse” mode) to those in Section IV of this work (where a fixed model-order of $L = 3$ was employed for the sparse mode), it is evident that the greedy scheme from [3] performs slightly better. However, that greedy scheme has a significantly higher complexity, mainly because the EM-GM-AMP algorithm in Table II must be re-run for each tested value of L . For this reason, we focus on the fixed model-order approach in the remainder of this work, and regard accurate *and fast* model-order-selection as a topic of future study.

F. EM-GM-AMP Summary and Demonstration

The fixed- L EM-GM-AMP⁵ algorithm developed in the previous sections is summarized in Table II. For EM-BG-AMP (as previously described in [2]), one would simply run EM-GM-AMP with $L = 1$.

To demonstrate EM-GM-AMP’s ability to learn the underlying signal distribution, Fig. 1 shows examples of the GM-modeled signal distributions learned by EM-GM-AMP in both “sparse” and “heavy-tailed” modes. To create the figure, we first constructed the true signal vector $\mathbf{x} \in \mathbb{R}^N$ using $N = 2000$ independent draws of the true distribution $p_X(\cdot)$ shown in each of the subplots. Then, we constructed the matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ and the noise vector $\mathbf{w} \in \mathbb{R}^M$ using independent zero-mean Gaussian draws, with $M = 1000$ and the noise variance adjusted to achieve SNR=25 dB in the resulting observations $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}$. Finally, we ran EM-GM-AMP according to Table II, and plotted the GM approximation $p_X(\mathbf{x}; \mathbf{q}^i)$ from (3) using the learned pdf parameters $\mathbf{q}^i = [\lambda^i, \omega^i, \theta^i, \phi^i, \psi^i]$. Figure 1 confirms that EM-GM-AMP is successful in learning a reasonable approximation of the unknown true pdf $p_X(\cdot)$ from the noisy compressed observations \mathbf{y} , in both sparse and heavy-tailed modes.

IV. NUMERICAL RESULTS

In this section we report the results of a detailed numerical study that investigate the performance of EM-GM-AMP under

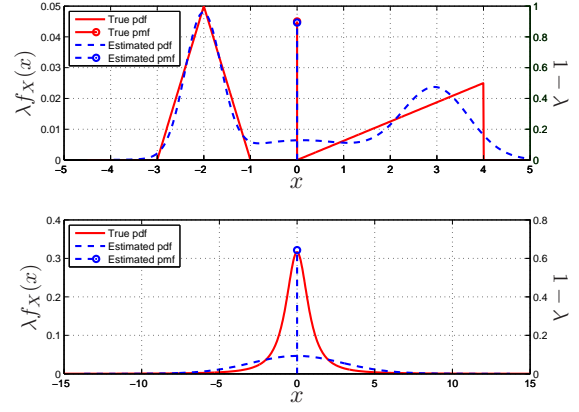


Fig. 1. True and EM-GM-AMP-learned versions of the signal distribution $p_X(\mathbf{x}) = \lambda f_X(\mathbf{x}) + (1 - \lambda)\delta(\mathbf{x})$. The top subplot shows “sparse” mode EM-GM-AMP run using GM-order $L = 3$ on a sparse signal whose non-zero components were generated according to a triangular mixture, whereas the bottom subplot shows “heavy-tailed” EM-GM-AMP run using $L = 4$ on a Student’s-t signal with rate parameter $q = 1.67$ (defined in (70)). The density of the continuous component $\lambda f_X(\mathbf{x})$ is marked on the left axis, while the mass of the discrete component $(1 - \lambda)\delta(\mathbf{x})$ is marked on the right axis.

```

Initialize  $L$  as described in Section III-E.
Initialize  $\mathbf{q}^0$  as described in Section III-D.
Initialize  $\hat{\mathbf{x}}^0 = \mathbf{0}$ .
for  $i = 1$  to  $I_{\max}$  do
    Generate  $\hat{\mathbf{x}}^i, \hat{\mathbf{z}}^i, (\mu^z)^i, \pi^i, \{\beta_k^i, \gamma_k^i, \nu_k^i\}_{k=1}^L$  using GM-GAMP
    with  $\mathbf{q}^{i-1}$  (see Table I).
    if  $\|\hat{\mathbf{x}}^i - \hat{\mathbf{x}}^{i-1}\|_2^2 < \tau_{\text{em}}\|\hat{\mathbf{x}}^{i-1}\|_2^2$  then
        break.
    end if
    Compute  $\lambda^i$  from  $\pi^{i-1}$  as described in (29).
    for  $k = 1$  to  $L$  do
        if sparse mode enabled then
            Compute  $\theta_k^i$  from  $\pi^{i-1}, \gamma_k^{i-1}, \{\beta_l^{i-1}\}_{l=1}^L$  as described in
            (50).
        else if heavy-tailed mode enabled then
            Set  $\theta_k^i = 0$ .
        end if
        Compute  $\phi_k^i$  from  $\theta_k^{i-1}, \pi^{i-1}, \gamma_k^{i-1}, \nu_k^{i-1}, \{\beta_l^{i-1}\}_{l=1}^L$  as
        described in (56).
        Compute  $\omega^i$  from  $\pi^{i-1}$  and  $\{\beta_l^{i-1}\}_{l=1}^L$  as described in (65).
    end for
    Compute  $\psi^i$  from  $\hat{\mathbf{z}}^i$  and  $(\mu^z)^i$  as in (23).6
end for

```

TABLE II
THE EM-GM-AMP ALGORITHM (FIXED- L CASE)

both noiseless and noisy settings. For all experiments, we set the GM-GAMP tolerance to $\tau_{\text{gamp}} = 10^{-5}$ and the maximum GAMP-iterations to $T_{\max} = 20$ (recall Table I), and we set the EM tolerance to $\tau_{\text{em}} = 10^{-5}$ and the maximum EM-iterations to $I_{\max} = 20$ (recall Table II). In “sparse” mode, we use GM order $L = 3$, while in “heavy-tailed” mode, we use $L = 4$.

A. Noiseless Phase Transitions

We first describe the results of experiments that computed noiseless empirical phase transition curves (PTCs) under three sparse-signal distributions. To evaluate each empirical PTC, we fixed $N = 1000$ and constructed a 30×30 grid where (M, K) were chosen to yield a uniform sampling of oversampling ratios $\frac{M}{N} \in [0.05, 0.95]$ and sparsity ratios $\frac{K}{M} \in [0.05, 0.95]$. At each grid point, we generated $R = 100$ independent realizations of a K -sparse signal \mathbf{x} from a spec-

⁵Matlab code available at <http://www.ece.osu.edu/~schniter/EMturboGAMP>.

⁶Empirically, we have observed that the EM update for ψ works better with the μ_m^z term in (23) weighted by $\frac{M}{N}$ and suppressed until later EM iterations. We conjecture that this is due to bias in the finite-sample GAMP variance estimates μ_m^z .

ified distribution and an $M \times N$ measurement matrix \mathbf{A} with i.i.d $\mathcal{N}(0, M^{-1})$ entries. From the noiseless measurements $\mathbf{y} = \mathbf{A}\mathbf{x}$, we recovered the signal \mathbf{x} using several algorithms. A recovery $\hat{\mathbf{x}}$ from realization $r \in \{1, \dots, R\}$ was defined a success if the $\text{NMSE} \triangleq \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 / \|\mathbf{x}\|_2^2 < 10^{-4}$, and the average success rate was defined as $\bar{S} \triangleq \frac{1}{R} \sum_{r=1}^R S_r$, where $S_r = 1$ for a success and $S_r = 0$ otherwise. The empirical PTC was then plotted, using Matlab's `contour` command, as the $\bar{S} = 0.5$ contour over the sparsity-undersampling grid.

Figures 2-4 show the empirical PTCs for four recovery algorithms: the proposed EM-GM-AMP algorithm (in “sparse” mode), the proposed EM-BG-AMP algorithm, a genie-tuned⁷ GM-AMP that uses the true parameters $\mathbf{q} = [\lambda, \omega, \theta, \phi, \psi]$, and the Donoho/Maleki/Montanari (DMM) Lasso-style AMP from [10]. For comparison, Figs. 2-4 also display the theoretical Lasso PTC (66). The signals were generated as Bernoulli-Gaussian (BG) in Fig. 2 (using mean $\theta = 0$ and variance $\phi = 1$ for the Gaussian component), as Bernoulli in Fig. 3 (i.e., all non-zero coefficients set equal to 1), and as Bernoulli-Rademacher (BR) in Fig. 4 (i.e., non-zero coefficients chosen uniformly at random from the set $\{-1, 1\}$).

For all three signal types, Figs. 2-4 show that the empirical PTC of EM-GM-AMP significantly improves on the empirical PTC of DMM-AMP as well as the theoretical PTC of Lasso. (The latter two are known to converge in the large system limit [10].) For BG signals, Fig. 2 shows that EM-GM-AMP and EM-BG-AMP both yield PTCs that are nearly identical to that of genie-GM-AMP, suggesting that our EM-learning procedure is working well. Note that, for these BG signals, the model employed by EM-BG-AMP is perfectly matched to the true signal, whereas that employed by EM-GM-AMP (with $L = 3$) has the potential to over-fit the true signal, although Fig. 2 suggests that this does not happen. For Bernoulli signals, Fig. 3 shows EM-BG-AMP performing nearly the same as genie-GM-AMP, and EM-GM-AMP performing even better than genie-GM-AMP. The latter behavior is due to EM-GM-AMP's ability to do per-realization parameter tuning, whereas genie-GM-AMP employs the best set of *fixed* parameters over all realizations. Finally, for BR signals, Fig. 4 shows EM-GM-AMP performing significantly better than EM-BG-AMP, since the former has the ability to accurately model the BR distribution (with $L \geq 2$ mixture components), whereas the latter (with a single mixture component) does not. Figure 4 again shows EM-GM-AMP performing slightly better than genie-GM-AMP in some regions of the sparsity-undersampling plane as a result of per-realization parameter tuning.

B. Noisy Sparse Signal Recovery

Figures 5-7 show NMSE for noisy recovery of BG, Bernoulli, and BR signals, respectively. To construct these plots, we fixed $N = 1000$, $K = 100$, $\text{SNR} = 25$ dB, and varied M . Each data point represents NMSE averaged over $R = 500$ realizations. For comparison, we show the performance of the proposed EM-GM-AMP (in “sparse”

⁷For genie-tuned GM-AMP, for numerical reasons, we set the noise variance at $\psi = 10^{-6}$ and, with Bernoulli and BR signals, the mixture variances at $\phi_k = 10^{-2}$.

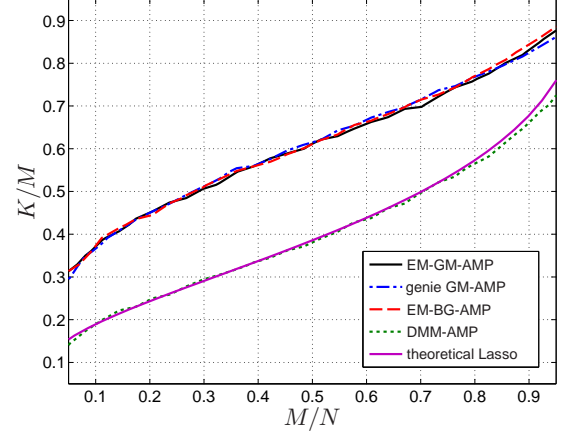


Fig. 2. Empirical PTCs and Lasso theoretical PTC for noiseless recovery of Bernoulli-Gaussian signals.

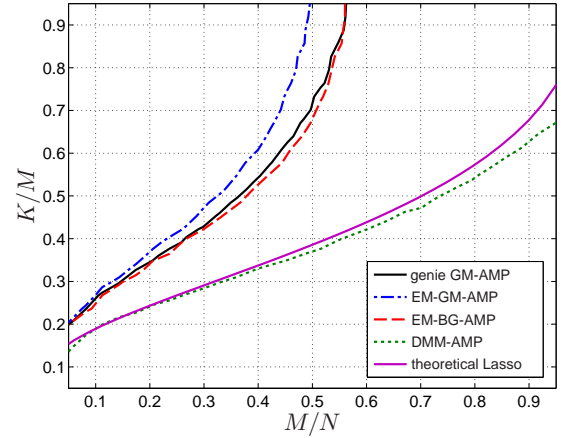


Fig. 3. Empirical PTCs and Lasso theoretical PTC for noiseless recovery of Bernoulli signals.

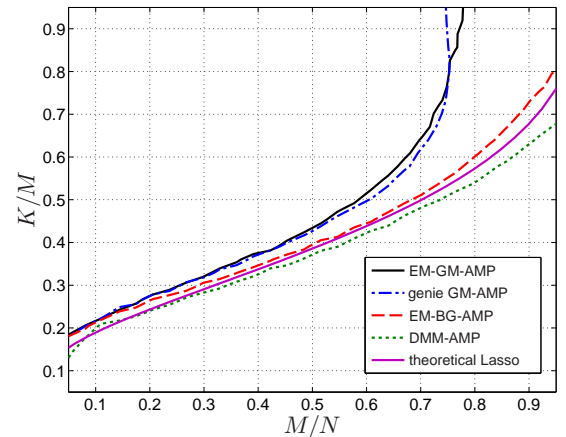


Fig. 4. Empirical PTCs and Lasso theoretical PTC for noiseless recovery of Bernoulli-Rademacher signals.

mode), EM-BG-AMP, genie-tuned⁸ Orthogonal Matching Pursuit (OMP) [24], genie-tuned⁸ Subspace Pursuit (SP) [25], Bayesian Compressive Sensing (BCS) [19], Sparse Bayesian Learning [18] (via the more robust T-MSBL [26]), de-biased

⁸We ran both OMP (using the implementation from <http://sparselab.stanford.edu/OptimalTuning/code.htm>) and SP under 10 different sparsity assumptions, spaced uniformly from 1 to $2K$, and reported the lowest NMSE among the results.

genie-tuned⁹ Lasso (via SPGL1 [27]), and Smoothed- ℓ_0 (SL0) [28]. All algorithms were run under the suggested defaults, with ‘noise=small’ in T-MSBL.

For BG signals, Fig. 5 shows that EM-GM-AMP and EM-BG-AMP together exhibit the best performance among the tested algorithms, reducing the M/N breakpoint (i.e., the location of the knee in the NMSE curve, which represents a sort of phase transition) from 0.3 down to 0.26, but also improving NMSE by ≈ 1 dB relative to the next best algorithm, which was BCS. For Bernoulli signals, Fig. 6 shows much more significant gains for EM-GM-AMP and EM-BG-AMP over the other algorithms: the M/N breakpoint was reduced from 0.4 down to 0.32, and the NMSE was reduced by ≈ 8 dB relative to the next best algorithm, which was T-MSBL in this case. Finally, for BR signals, Fig. 7 shows a distinct advantage for EM-GM-AMP over the other algorithms, including EM-BG-AMP, due to the former’s unique ability to accurately model the BR signal prior. In particular, for $M/N \geq 0.36$, EM-GM-AMP reduces the NMSE by at least 8 dB relative to the best of the other algorithms (which was either EM-BG-AMP or T-MSBL depending on the value of M/N) and reduces the M/N breakpoint from 0.38 down to 0.36.

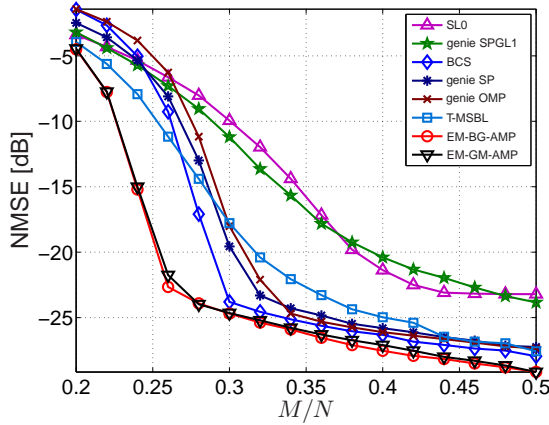


Fig. 5. NMSE versus undersampling ratio M/N for noisy recovery of Bernoulli-Gaussian signals.

To investigate each algorithm’s robustness to AWGN, we plotted the NMSE attained in the recovery of BR signals with $N = 1000$, $M = 500$, and $K = 100$ as a function of SNR in Fig. 8, where each point represents an average over $R = 100$ problem realizations. All algorithms were under the same conditions as those reported previously, except that T-MSBL used ‘noise=small’ when $\text{SNR} > 22$ dB and ‘noise=mild’ when $\text{SNR} \leq 22$ dB, as recommended in [29]. From Fig. 8, we see that the essential behavior observed in the fixed-SNR BR plot Fig. 7 holds over a wide range of SNRs. In particular, Fig. 8 shows that EM-GM-AMP yields significantly lower NMSE than all other algorithms over the full SNR range, while EM-BG-AMP and T-MSBL yield the second lowest NMSE (also matched by BCS for SNRs between 30 and 40 dB). Note, however, that T-MSBL must be

⁹We ran SPGL1 in ‘BPDN’ mode: $\min_{\mathbf{x}} \|\mathbf{x}\|_1$ s.t. $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \sigma$, for hypothesized tolerances $\sigma^2 \in \{0.1, 0.2, \dots, 1.5\} \times M\psi$, and reported the lowest NMSE among the results.

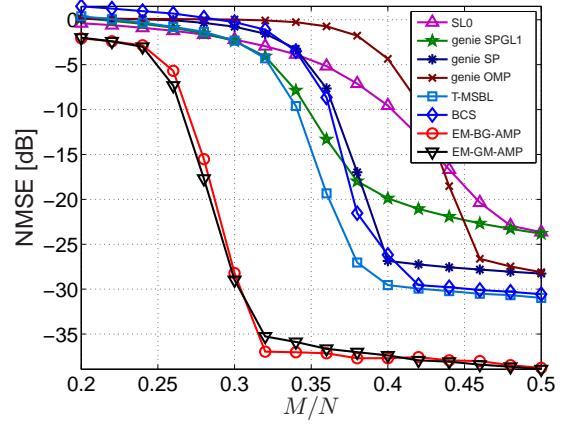


Fig. 6. NMSE versus undersampling ratio M/N for noisy recovery of Bernoulli signals.

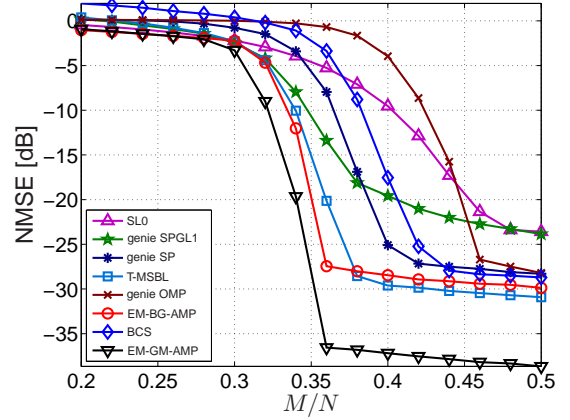


Fig. 7. NMSE versus undersampling ratio M/N for noisy recovery of Bernoulli-Rademacher signals.

given some knowledge about the true noise variance in order to perform well [29], unlike EM-GM-AMP or EM-BG-AMP.

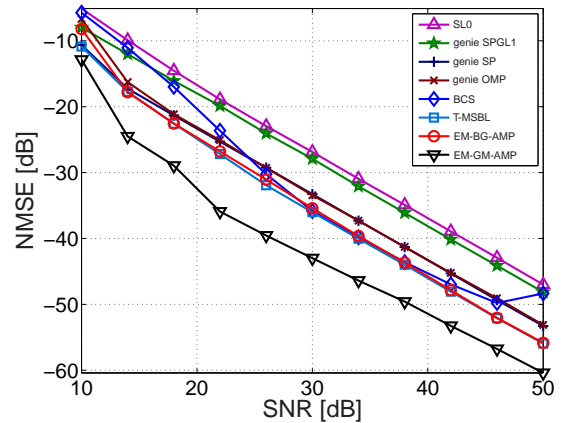


Fig. 8. NMSE versus SNR for noisy recovery of Bernoulli-Rademacher signals.

C. Heavy-Tailed Signal Recovery

In many applications of compressive sensing, the signal to be recovered is not perfectly sparse, but rather “heavy tailed,” in that there are a few large coefficients and many small (but nonzero) ones. To investigate algorithm performance in this

setting, we constructed the signal vector \mathbf{x} as i.i.d Student's-t, i.e., with prior pdf

$$p_X(x; q) \triangleq \frac{\Gamma((q+1)/2)}{\sqrt{2\pi}\Gamma(q/2)} (1+x^2)^{-(q+1)/2} \quad (70)$$

under the (non-compressible) rate $q = 1.67$, which has been shown to be an excellent model for wavelet coefficients of natural images [30]. For such signals, Fig. 9 plots NMSE versus the number of measurements M for fixed $N = 1000$, $\text{SNR} = 25$ dB, and an average of $R = 500$ realizations. Figure 9 shows EM-GM-AMP (here run in “heavy-tailed” mode) outperforming all other algorithms under test.¹⁰ We have also verified (in experiments not shown here) that “heavy-tailed” EM-GM-AMP exhibits similarly good performance with other values of the Student's-t rate parameter q .

It may be interesting to notice that, with the perfectly sparse signals examined in Figs. 5-7, the mixed-norm approaches (i.e., SL0 and SPGL1) performed relatively poorly, the relevance-vector-machine (RVM)-based approaches (i.e., BCS, T-MSBL) performed relatively well, and the greedy approaches (OMP and SP) performed in-between. Meanwhile, with the heavy-tailed signals in Fig. 9, the situation was reversed: the mixed-norm approaches performed relatively well, whereas the RVM approaches performed relatively poorly. Thus, it appears that the mixed-norm approaches are somehow better “tuned” to heavy-tailed signals, whereas the RVM approaches are better tuned to sparse signals. For all signal types, though, the best recovery performance came from EM-GM-AMP, and we attribute this behavior to EM-GM-AMP's ability to tune itself to the signal realization at hand.

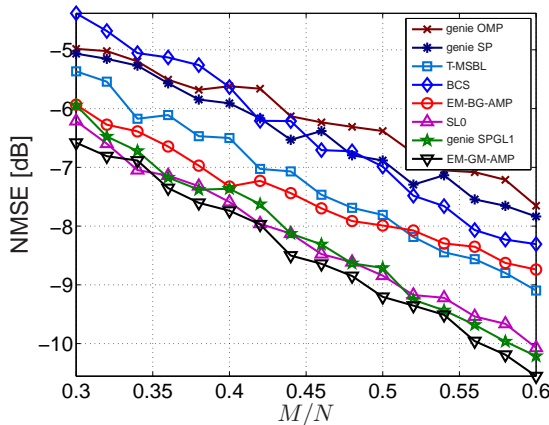


Fig. 9. NMSE versus undersampling ratio M/N for noisy recovery of Student's-t signals.

D. Runtime and Complexity Scaling with N

Next we investigated how complexity scales with signal length N by evaluating the runtime of each algorithm on a typical personal computer. For this, we fixed $K/N = 0.1$, $M/N = 0.5$, $\text{SNR} = 25$ dB and varied the signal length N . Figure 10 shows the runtimes for noisy recovery of a

¹⁰In this experiment, we ran both OMP and SP under 10 different sparsity hypotheses, spaced uniformly from 1 to $K_{\text{lasso}} = M\rho_{\text{SE}}(\frac{M}{N})$, and reported the lowest NMSE among the results.

Bernoulli-Rademacher signal, while Fig. 11 shows the corresponding NMSEs. In these plots, each datapoint represents an average over $R = 50$ realizations. The algorithms that we tested are the same ones that we described earlier. However, to fairly evaluate runtime, we configured some a bit differently than before. In particular, for genie-tuned SPGL1, in order to yield a better runtime-vs-NMSE tradeoff, we reduced the tolerance grid (recall footnote 9) to $\sigma^2 \in \{0.6, 0.8, \dots, 1.4\} \times M\psi$ and turned off debiasing. For OMP and SP, we used under the fixed support size $K_{\text{lasso}} = M\rho_{\text{SE}}(\frac{M}{N})$ rather than searching for the size that minimizes NMSE over a grid of 10 hypotheses, as before. Otherwise, all algorithms were run under the suggested defaults, with T-MSBL run under ‘noise=small’ and EM-GM-AMP run in “sparse” mode.

The complexities of EM-BG-AMP and EM-GM-AMP are dominated by one matrix multiplication by \mathbf{A} and \mathbf{A}^\top per iteration, and the number of iterations is invariant to M and N . Thus, when these matrix multiplications are explicitly implemented and \mathbf{A} is dense, the total complexity of EM-BG-AMP and EM-GM-AMP should scale as $\mathcal{O}(MN)$. This scaling is indeed visible in the runtime curves of Fig. 10. There, $\mathcal{O}(MN)$ becomes $\mathcal{O}(N^2)$ since the ratio M/N was fixed, and the horizontal axis plots N on a logarithmic scale, so that this complexity scaling manifests, at sufficiently large values of N , as a line with slope 2. Figure 10 confirms that genie-tuned SPGL1 also has the same complexity scaling, albeit with longer overall runtimes. Meanwhile, Fig. 10 shows T-MSBL, BCS, SL0, OMP, and SP exhibiting a complexity scaling of $\mathcal{O}(N^3)$ (under fixed K/N and M/N), which results in orders-of-magnitude larger runtimes for long signals (e.g., $N \geq 10^4$). With short signals (e.g., $N < 1300$), though, OMP, SP, SL0, and SPGL1 are faster than EM-GM-AMP. Finally, Fig. 11 verifies that, for most of the algorithms, the NMSEs are relatively insensitive to signal length N when the undersampling ratio M/N and sparsity ratio K/M are both fixed, although the performance of EM-GM-AMP improves with N (which is not surprising in light of AMP's large-system-limit optimality properties [13]) and the performance of BCS degrades with N .

The proposed EM-GM-AMP and EM-BG-AMP algorithms, as well as SPGL1, can also handle the case where multiplication by \mathbf{A} and \mathbf{A}^\top is implemented using a fast algorithm like the fast Fourier transform (FFT)¹¹, which reduces the complexity to $\mathcal{O}(N \log N)$, and avoids the need to store \mathbf{A} in memory—a serious problem when MN is large. The dashed lines in Figs. 10-11 (labeled “fft”) show the average runtime and NMSE of EM-GM-AMP, EM-BG-AMP, and SPGL1 in case that \mathbf{A} was a randomly row-sampled FFT. As expected, the runtimes are dramatically reduced. While EM-BG-AMP retains its place as the fastest algorithm, SPGL1 now runs $1.5\times$ faster than EM-GM-AMP (at the cost of 14 dB higher NMSE).

¹¹For our FFT-based experiments, we used the complex-valued versions of EM-BG-AMP, EM-GM-AMP, and SPGL1.

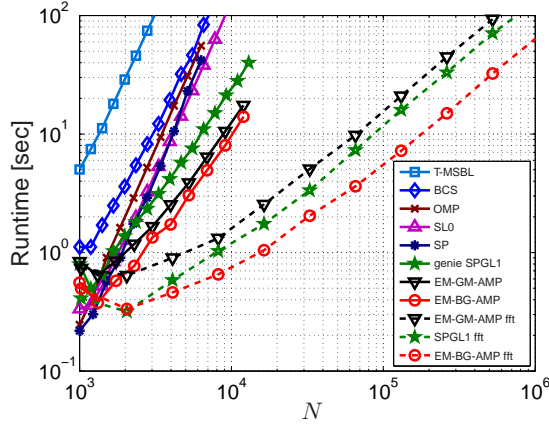


Fig. 10. Runtime versus signal length N for noisy recovery of Bernoulli-Rademacher signals.

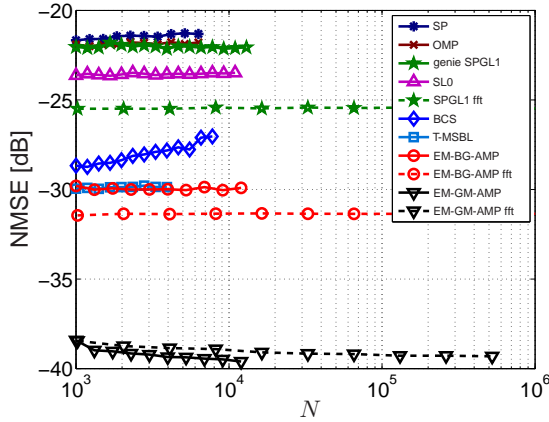


Fig. 11. NMSE versus signal length N for noisy recovery of Bernoulli-Rademacher signals.

E. Example: Compressive Recovery of Audio

As a practical example, we experimented with the recovery of an audio signal from compressed measurements. The full length-81920 audio signal was first partitioned into T blocks $\{\mathbf{u}_t\}_{t=1}^T$ of length N . Noiseless compressed measurements $\mathbf{y}_t = \Phi \mathbf{u}_t \in \mathbb{R}^M$ were then collected using $M = N/2$ samples per block. Rather than reconstructing \mathbf{u}_t directly from \mathbf{y}_t , we first reconstructed¹² the transform coefficients $\mathbf{x}_t = \Psi^T \mathbf{u}_t$, using the (orthogonal) discrete cosine transform (DCT) $\Psi \in \mathbb{R}^{N \times N}$, and later reconstructed \mathbf{u}_t via $\mathbf{u}_t = \Psi \mathbf{x}_t$. Our effective sparse-signal model can thus be written as $\mathbf{y}_t = \mathbf{A} \mathbf{x}_t$ with $\mathbf{A} = \Phi \Psi$. We experimented with two types of measurement matrix Φ : i.i.d Gaussian and random selection (i.e., containing rows of the identity matrix selected uniformly at random), noting that the latter allows a fast implementation of \mathbf{A} and \mathbf{A}^T . Table III shows the resulting time-averaged NMSE, i.e., $\text{TNMSE} \triangleq \frac{1}{T} \sum_{t=1}^T \|\mathbf{u}_t - \hat{\mathbf{u}}_t\|^2 / \|\mathbf{u}_t\|^2$, and total runtime achieved by the previously described algorithms at block lengths $N = 1024, 2048, 4096, 8192$, which correspond to $T = 80, 40, 20, 10$ blocks, respectively. The numbers

¹²Although one could exploit additional structure among the multiple-timestep coefficients $\{\mathbf{x}_t\}_{t=1}^T$ for improved recovery (e.g., sparsity clustering in the time and/or frequency dimensions, as well as amplitude correlation in those dimensions) as demonstrated in [31], such techniques are outside the scope of this paper.

reported in the table represent an average over 50 realizations of Φ . For these experiments, we configured the algorithms as described in Section IV-C for the heavy-tailed experiment except that, for genie-SPGL1, rather than using $\psi = 0$, we used $\psi = 10^{-6}$ for the tolerance grid (recall footnote 9) because we found that this value minimized TNMSE and, for T-MSBL, we used the setting `prune_gamma` = 10^{-12} as recommended in a personal correspondence with the author. For certain combinations of algorithm and blocklength, excessive runtimes prevented us from carrying out the experiment, and thus no result appears in the table.

Table III shows that, for this audio experiment, EM-GM-AMP and SL0 performed best in terms of TNMSE, and EM-BG-AMP performed second best. As in the synthetic examples presented earlier, we attribute EM-GM-AMP's excellent TNMSE to its ability to tune itself to whatever signal is at hand. As for SL0's excellent TNMSE, we reason that it had the good fortune of being particularly well-tuned to this audio signal, given that it performed relatively poorly with the signal types used for Figs. 5-8. From the runtimes reported in Table III, we see that, with i.i.d Gaussian Φ and the shortest block length ($N = 1024$), OMP is by far the fastest, whereas EM-BG-AMP and EM-GM-AMP are the slowest. But, as the block length grows, EM-BG-AMP and EM-GM-AMP achieve better and better runtimes as a consequence of their excellent complexity scaling, and eventually become the fastest of the algorithms under test (as shown with i.i.d Gaussian Φ at $N = 8192$). For this audio example, the large-block regime may be the more important, because that is where all algorithms give smallest TNMSE. Next, looking at the runtimes under random-selection Φ , we see dramatic speed improvements for EM-GM-AMP, EM-BG-AMP, and SPGL1, which were all able to leverage Matlab's fast DCT. In fact, the total runtimes of these three algorithms *decrease* as N is increased from 1024 to 8192. We conclude by noting that EM-BG-AMP (at $N = 8192$ with random selection Φ) achieves the fastest runtime in the entire table while yielding a TNMSE that is within 1 dB of the best value in the entire table. Meanwhile, EM-GM-AMP (at $N = 8192$ with random selection Φ) yields the best TNMSE in the entire table while taking only about twice as long to run as the fastest time in the entire table.

V. CONCLUSIONS

Those interested in practical compressive sensing face the daunting task of choosing among literally hundreds of signal reconstruction algorithms (see, e.g., [32]). In testing these algorithms, they are likely to find that some work very well with particular signal classes, but not with others. They are also likely to get frustrated by those algorithms that require the tuning of many parameters. Finally, they are likely to find that some of the algorithms that are commonly regarded as “very fast” are actually very slow in high-dimensional problems. Meanwhile, those familiar with the theory of compressive sensing know that the workhorse Lasso is nearly minimax optimal, and that its phase transition curve is robust to the nonzero-coefficient distribution of sparse signals. However,

		$N = 1024$		$N = 2048$		$N = 4096$		$N = 8192$	
		TN MSE	time	TN MSE	time	TN MSE	time	TN MSE	time
i.i.d. Gaussian Φ	EM-GM-AMP	-16.9	159.2	-18.0	213.2	-20.7	434.0	-21.4	1129
	EM-BG-AMP	-15.9	115.2	-17.0	174.1	-19.4	430.2	-20.0	1116
	SL0	-16.8	41.6	-17.9	128.5	-20.6	629.0	-21.3	2739
	genie SPGL1	-14.3	90.9	-16.2	200.6	-18.6	514.3	-19.5	1568
	BCS	-15.0	67.5	-15.8	149.1	-18.4	428.0	-18.8	2295
	SBL	-16.3	1.2e4	—	—	—	—	—	—
	genie OMP	-13.9	20.1	-14.9	109.9	-17.6	527.0	—	—
	genie SP	-14.5	87.7	-15.5	305.9	-18.0	1331	—	—
random selection Φ	EM-GM-AMP	-16.7	56.1	-17.7	43.7	-20.5	38.0	-21.5	37.8
	EM-BG-AMP	-16.2	29.6	-17.2	22.3	-19.7	19.4	-20.5	18.0
	SL0	-16.7	35.7	-17.6	119.5	-20.4	597.8	-21.2	2739
	genie SPGL1	-14.0	34.4	-15.9	24.5	-18.4	21.7	-19.7	19.6
	BCS	-15.5	60.5	-16.1	126.2	-19.4	373.8	-20.2	2295
	SBL	-15.5	1.2e4	—	—	—	—	—	—
	genie OMP	-15.1	20.1	-15.7	106.8	-18.9	506.0	—	—
	genie SP	-15.2	104.5	-16.1	395.3	-18.7	1808	—	—

TABLE III
AVERAGE TN MSE (IN DB) AND TOTAL RUNTIME (IN SECONDS) FOR
COMPRESSIVE AUDIO RECOVERY.

they also know that, for most signal classes, there is a large gap between the MSE performance of Lasso and that of the MMSE estimator derived under full knowledge of the signal and noise statistics [11]. Thus, they may wonder whether there is a way to close this gap by designing a signal reconstruction algorithm that *both learns and exploits* the signal and noise statistics.

With these considerations in mind, we proposed an empirical Bayesian approach to compressive signal recovery that merges two powerful inference frameworks: expectation maximization (EM) and approximate message passing (AMP). We then demonstrated—through a detailed numerical study—that our approach, when used with a flexible Gaussian-mixture signal prior, achieves a state-of-the-art combination of reconstruction error and runtime on a very wide range of signal types in the high-dimensional regime.

REFERENCES

- [1] J. P. Vila and P. Schniter, “An empirical-Bayes approach to compressive sensing via approximate message passing,” presented at the *Duke Workshop on Sensing and Analysis of High-Dimensional Data*, (Durham, NC), July 2011.
- [2] J. P. Vila and P. Schniter, “Expectation-maximization Bernoulli-Gaussian approximate message passing,” in *Proc. Asilomar Conf. Signals Syst. Comput.*, (Pacific Grove, CA), pp. 799–803, Nov. 2011.
- [3] J. P. Vila and P. Schniter, “Expectation-maximization Gaussian-mixture approximate message passing,” in *Proc. Conf. Inform. Science & Syst.*, (Princeton, NJ), pp. 1–6, Mar. 2012.
- [4] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. New York: Cambridge Univ. Press, 2012.
- [5] M. Bayati, M. Lelarge, and A. Montanari, “Universality in polytope phase transitions and iterative algorithms,” in *Proc. IEEE Int. Symp. Inform. Thy.*, (Boston, Ma), pp. 1–5, June 2012.
- [6] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *J. Roy. Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.
- [7] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit,” *SIAM J. Scientific Comput.*, vol. 20, no. 1, pp. 33–61, 1998.
- [8] D. L. Donoho and J. Tanner, “Observed universality of phase transitions in high-dimensional geometry, with implications for modern data analysis and signal processing,” *Phil. Trans. Royal Soc. A*, vol. 367, no. 1906, pp. 4273–4293, 2009.
- [9] D. L. Donoho, A. Maleki, and A. Montanari, “The noise-sensitivity phase transition in compressed sensing,” *arXiv:1004.1218*, Apr. 2010.
- [10] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing,” *Proc. Nat. Acad. Sci.*, vol. 106, pp. 18914–18919, Nov. 2009.

- [11] Y. Wu and S. Verdú, “Optimal phase transitions in compressed sensing,” *arXiv:1111.6822*, Nov. 2011.
- [12] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing: I. Motivation and construction,” in *Proc. Inform. Theory Workshop*, (Cairo, Egypt), pp. 1–5, Jan. 2010.
- [13] M. Bayati and A. Montanari, “The dynamics of message passing on dense graphs, with applications to compressed sensing,” *IEEE Trans. Inform. Theory*, vol. 57, pp. 764–785, Feb. 2011.
- [14] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” *arXiv:1010.5141*, Oct. 2010.
- [15] A. Dempster, N. M. Laird, and D. B. Rubin, “Maximum-likelihood from incomplete data via the EM algorithm,” *J. Roy. Statist. Soc.*, vol. 39, pp. 1–17, 1977.
- [16] B. Efron, *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. New York: Cambridge University Press, 2010.
- [17] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.
- [18] D. P. Wipf and B. D. Rao, “Sparse Bayesian learning for basis selection,” *IEEE Trans. Signal Process.*, vol. 52, pp. 2153–2164, Aug. 2004.
- [19] S. Ji, Y. Xue, and L. Carin, “Bayesian compressive sensing,” *IEEE Trans. Signal Process.*, vol. 56, pp. 2346–2356, June 2008.
- [20] R. Neal and G. Hinton, “A view of the EM algorithm that justifies incremental, sparse, and other variants,” in *Learning in Graphical Models* (M. I. Jordan, ed.), pp. 355–368, MIT Press, 1999.
- [21] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2007.
- [22] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, “Statistical physics-based reconstruction in compressed sensing,” *arXiv:1109.4424*, Sept. 2011.
- [23] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, “SMEM algorithm for mixture models,” *Neural Comput.*, vol. 12, pp. 2109–2128, Sept. 2000.
- [24] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Proc. 27th Ann. Asilomar Conf. Signals, Systems, and Computers*, 1993.
- [25] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing reconstruction,” *IEEE Trans. Inform. Theory*, vol. 55, pp. 2230–2249, Mar. 2009.
- [26] Z. Zhang and B. D. Rao, “Sparse signal recovery with temporally correlated source vectors using sparse Bayesian learning,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, pp. 912–926, Sept. 2011.
- [27] E. van den Berg and M. P. Friedlander, “Probing the Pareto frontier for basis pursuit solutions,” *SIAM J. Scientific Comput.*, vol. 31, no. 2, pp. 890–912, 2008.
- [28] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, “A fast approach for overcomplete sparse decomposition based on smoothed norm,” *IEEE Trans. Signal Process.*, vol. 57, pp. 289–301, Jan. 2009.
- [29] Z. Zhang, “Master the usage of T-MSBL in 3 minutes,” tech. rep., Univ. of California, San Diego, Nov. 2011.
- [30] V. Cevher, “Learning with compressible priors,” in *Proc. Neural Inform. Process. Syst. Conf.*, (Vancouver, B.C.), pp. 261–269, Dec. 2009.
- [31] J. Ziniel, S. Rangan, and P. Schniter, “A generalized framework for learning and recovery of structured sparse signals,” in *Proc. IEEE Workshop Statist. Signal Process.*, (Ann Arbor, MI), Aug. 2012.
- [32] “Compressive sensing resources: References and software,” <http://dsp.rice.edu/cs>.